14
From Pattern Dictionary to Patternbank

Elisabetta Jezek & Francesca Frontini

Abstract

In this paper we focus on the tension between the semantic types (STs) associated with verb arguments and their extensional definition, i.e. the lexical set (LS) that may fill the different argument positions – a tension that the work within the Pattern Dictionary of English Verbs (PDEV, Hanks 2007) project coordinated by Patrick Hanks substantially contributed to identify, sharpen and problematize.

After reviewing Patrick's insights on this phenomenon, we propose that the analysis of the mismatch between STs and LSs aimed at building a corpus-based ontology for Word Sense Disambiguation (WSD, Hanks *et al*. 2007) can be improved by extending the Corpus Pattern Analysis (CPA, Hanks 2004) technique used in PDEV, so that it includes the annotation of verb patterns onto the corpus instances that instantiate them. This produces a resource (the "Patternbank") that not only allows one to see the patterns of each verb and to retrieve the relevant contexts (as in the initial PDEV architecture), but also to see how the elements of the patterns map specifically onto the elements of the context. A closer look at the benefits of pattern annotation reveals that it can be useful to single out linguistic phenomena pertaining not only to the semantics/ontology interface but also to the semantics/syntax interface (syntactic alternations, argument dropping), as well as for several NLP applications.

We report on the first steps taken in the planning of a "Patternbank" for Italian.

1. Introduction

One of the many ways in which the work of Patrick Hanks has contributed to our understanding of the organization of lexical knowledge (in particular, the distinction between dictionary and encyclopaedia) and of the processes underlying meaning modulation in language, is the study of the complex relation between the ontological classification proposed for words and their distributional/syntagmatic behaviour. More specifically, how types actually behave in context and how this behaviour can be modelled in a type system that is consistent with the conceptual organization unveiled by language use. For example, the analysis conducted with the aim of creating a repository of corpus-derived semantically motivated syntagmatic patterns for verbs for purposes of Word Sense Disambiguation (WSD),[1] has clearly shown that argument types co-determine verb meaning in context and can therefore be used predictably to induce what meaning a verb is likely to exhibit given a certain semantic environment:

> *fire*
> if [[Weapon]-obj] then => 'cause to discharge a projectile'
> if [[Human]-obj] then => 'dismiss from employment'[2]
> (Hanks & Pustejovsky 2005)

Equally important are the problems that this type of research has contributed to identify. For example, it has highlighted that two main orders of problems arise when semantic types and ontologies are used to perform WSD.

The first order of problem touches the foundations of ontological representation and the complex interplay between semantics and cognition (Jackendoff 2002: 267-291). Which type distinctions are relevant for disambiguating word meaning in context (i.e. are linguistically relevant) and how can those be identified operationally? Which granularity should those distinctions have? As noted in Hanks *et al*. (2007), current ontologies used for disambiguation often include conceptual nodes that are inappropriate (or even misleading) for the task they are supposed to perform. For example, WordNet's hierarchies introduce a large number of scientifically motivated fine-grained subdivisions (e.g. <chordate>) that do not reflect ordinary language use and are useless in WSD tasks. According to Patrick, this problem is not an accidental fact related to the contingencies of a given ontology. Rather, it is the result of the overall procedure through which ontologies of this kind are built. Canonically, conceptual distinctions in these type systems are not grounded in the analysis of real corpus data. As a result, the connection between the taxonomic representation proposed for words and their distributional behaviour is poor, and the performance of such resources in WSD tasks is low.

A second order of problem touches on the interplay between the semantic types (STs) associated with verb arguments and their extensional definition, i.e. the paradigmatic group of words that may fill the argument positions in a pattern (the *lexical set* or LS in Hanks's 1996 terminology). The work of Patrick and his collaborators has substantially contributed to clarify two major sub-problems of this interaction. These sub-problems stand in the way of fulfilling the goal of grouping lexical sets into a hierarchical ontology in order to predict the meaning of verbs according to their context.

Sub-problem 1. Some words that instantiate a given argument type in context may not belong to the expected type (i.e. to the type required by the selecting verb), while other words that are potential candidates for the lexical set are not instantiated. In Patrick's words:

> Typically, a person "attends" an [Event] (meeting, lecture, funeral, coronation, etc.). However, there are many events (e.g. thunderstorm, suicide) that people do not "attend", while some of the things that people do attend (e.g. school, church, clinic) are not [Event]s, but rather [Location]s.
> (Hanks & Jezek 2008: 394)[3]

One possible explanation for the mismatch between LSs and STs observed in the data might be that types are modulated in context. In other words, the interaction between the meanings of the verb and that of the noun may cause the semantic type of N to be "stretched" or "shrunk" in composition, so that for a given verb type selectional requirement, no precise prediction seems possible as to what words make up the corresponding lexical set in context.[4] *Coercion* (or *type shifting*)[5] is a well studied phenomenon of the lexical set being bigger than the corresponding semantic type. Yet the reverse phenomenon (lexical set being smaller than the corresponding type) is probably as insidious and it suggests that the notion of type is not sufficient to account for the semantic behaviour of words in composition.

Sub-problem 2. Lexical sets are not stable paradigmatic structures. Instead, they *shimmer*, meaning that their internal composition changes from verb to verb, so that some items drop out and others come in according to context. In other words, the lexical set realizing a specific semantic type does not seem to be directly "transferable" to different verbs selecting the same type (except for some prototypical members, see Rumshisky 2008), an outcome which is both linguistically intriguing and computationally problematic. Hanks & Jezek (2008) exemplify the problem by using the semantic type [[Document]], which is selected by a number of verbs, like *read*, *publish*, *send* and *translate*. Apparently one could say that, except for cases of coercion such as [[Author of the Document]]=>[[Document]] all these verbs should display similar lexical sets. The analysis of corpus evidence shows, however, that this is not the case:

> What is a [[Document]]?
> *read* {book, newspaper, bible, article, letter, poem, novel, text, page, passage, story, comics script, poetry, report, page, label, verse, manual}
> *publish* {report, book, newspaper, article, pamphlet, edition, booklet, result, poem, document, leaflet, newsletter, volume, treatise, catalogue, findings, guide, novel, handbook, list}
> *send* {message, letter, telegram, copy, postcard, cheque, parcel, fax, card, document, invoice, mail, memo, report}
> *translate* {bible, text, instructions, abstract, treatise, book, document, extract, poem, menu, term, novel, message, letter}
> (Hanks & Jezek 2008: 399)

These data suggest that the mapping between the categories that make up our conceptual system and the selectional constraints that verbs impose on their arguments might not be straightforward. While in some cases the selectional requirement imposed by a predicate may correspond neatly to a category in our conceptual system, in other cases it may “cut” that category out of our conceptual system in a way which is consistent with the predication being made, though not necessarily corresponding to a pre-defined type.

This point becomes even more apparent if we consider the lexical extension of the direct object position of the basic sense of the English verbs *throw*, *carry* or *lift* (selected type [[Physical Object]]). Corpus evidence shows that the physical objects that one typically *throws*, *carries* or *lifts* do not overlap. While one typically *carries* or *lifts* a *suitcase*, one does not normally *throw* it; typically, one *lifts* a [[Body Part]] (*head*, *leg*, *hand*, *finger*, *arm*, *eyebrow*, *shoulder*, etc.) but one cannot *carry* or *throw* it; etc. From this evidence it is not necessary to derive that the three lexical sets constitute three distinct conceptual categories (i.e. subtypes of physical objects: [[Things that one can throw]], [[Things that one can carry]], etc.). Rather, one could assume that the three lexical sets become cognitively and linguistically relevant as classes by virtue of appearing in the same predicative context (i.e. of being “predicated” by the same verb).[6]

Also, types are canonically defined primarily on the basis of their Formal role, and organized in hierarchical ontologies accordingly.[7] It is evident, however, that other dimensions of meaning (for example, the Telic or the Constitutive dimension, that are orthogonal to the Formal axis) are relevant for ontological classification and ontology building.[8] It is well known that [[Parts]] (Constitutive role) require a different organizing principle than the IS-A relation expressed through the Formal: a *human hand* is a [[Living entity]], but it isn't [[Human]]. Furthermore, from a compositional perspective, it is evident that verb selection may point at attributes of objects that are more granular than types (and different from the Formal) and that the semantic behaviour of types in context may be determined/conditioned accordingly.[9]

Given the situation outlined above, the solution proposed by Patrick is radical: in order to model verb polysemy in a way which is computationally tractable (i.e. suitable to perform sense disambiguation successfully), we need a new type of ontology for nouns. This ontology must be truly *linguistic*, i.e. reflect the distinctions that prove to be relevant for language. According to Patrick, an ontology of this kind displays a number of properties:

(a) It is human-centred. As noted in Jackendoff (2002), inter alia, the part of cognition that is linguistically relevant seems to be characterized by being human-centred. That is, the distinctions that are relevant for language are those that are relevant for human beings. The more something is relevant for humans, the subtler it is categorized. This is confirmed by the fact that the majority of verbs require a human subject: 'Because language is anthropocentric, [[Human]] is by far the most frequent and the most important semantic type' (Hanks *et al.* 2007). Subsequently, the linguistic ontology does not display the same granularity for the whole semantic space. Its shape turns out to be lopsided and asymmetrical, for example: 'it necessarily devotes more attention to activities in which humans participate' (Hanks, Jezek & Lenci 2008);

(b) It is data-driven. The ontology that is necessary to model verb polysemy in language and that can subsequently be used successfully for disambiguation must be derived from the data (i.e. from our linguistic behaviour) and not be defined a priori. We must recognize the semantic oppositions that are relevant for human beings following a bottom-up procedure, using large corpora as the main source of evidence. Types must be drawn from the observation of the syntagmatic behaviour of words in actual usage. The organization of the lexicon in terms of semantic types must be derived from how we use words in context;

(c) It is shallow. In the linguistic ontology, types are sufficiently fine-grained to capture sense distinctions, so that a displacement in the ontology corresponds to a change in meaning in the co-occurring verbs. At the same time, they are general enough to be used in multiple contexts. As we saw above, it is often impossible to find a semantic type that matches all and only the lexical elements that can fill an argument position in a pattern. The chosen semantic type is often a compromise between accuracy and usability. Defining a semantic type that can work only with one verb would be missing the point of compositionality;

(d) Semantic types are defined extensionally in the form of lexical sets. The structure of the linguistic ontology is borne out of the analysis of the dynamic interaction between the two. Deriving STs from the corpus improves the matching between LSs and STs, but does not eliminate the problem. However, what is important is that the distinction between ST and LS is not flattened out, just as it happens in many quantitative studies on annotated corpora;

(e) It includes statistically relevant information on lexical sets. Each canonical member L of a lexical set is recorded with statistical contextual information on the total number of occurrences of L with V and the total number of occurrences of V in the reference corpus (cf. Hanks & Jezek 2008);

(f) It incorporates rules that account for the modulation of types observed in the data. Non-canonical lexical items of the lexical set are coerced into ‘honorary’ membership of a semantic type in particular context and classified as exploitations of the norms of usage (cf. the notion of *promoted literal type* in Pustejovsky, Hanks & Rumshisky 2004).

2. Extending CPA: From Pattern Dictionary to Patternbank

According to the theoretical issues raised in the previous section, logical steps for the implementation of a corpus-based ontology of Nouns for WSD are: by looking at the corpus, (a) define which verb patterns there are and which STs they define; (b) validate STs through careful examination of the LSs that instantiate them; and (c) organize the corpus-derived STs hierarchically to account for lexical inheritance.

Given the tension between STs and LSs alluded to in Section 1, one might expect to find a certain amount of mismatch in the data, partly due to the fact that the interaction between the meanings of the verb and that of the noun causes the semantic type of the noun to be “stretched” or “shrunk” in context. At the same time one should verify whether the STs defined in (a) actually identify the core of the instances for the patterns in which they are selected. In other words, if the type [[Measure Unit]] is identified, one expects to find a set of verb patterns that select [[Measure Unit]] as an argument; at the same time the LSs derived from the corpus for this argument in all those verbs should contain a significant and prototypical group of common elements.

Ideally, in order to perform this procedure successfully, the corpus analyst should be able to retrieve all possible combinations of LSs and STs. Given a verb V, a sense S, a semantic type T, an argument position A, and a lexical element L, we can redefine a pattern as a relation that connects {V, S, T, A}. In order to define corpus-based STs for verb patterns and organize them in an ontology, the corpus analyst would then want to:

- query for all L that fill {V, S, T, A}, to generate the LSs for a given pattern argument; for instance query all nouns that can fill the pattern [[Human]] *divora* [[Food]], where the sense of *divorare* ‘devour’ is that of “ingest and consume food hungrily”;
- compare all L that fill {V, S1, T1, A} with all L that fill {V, S2, T2, A} to verify whether some elements are common to more than one sense; for example, given the verb *finire* ‘finish’ with sense 1 ‘bring to an end, complete an activity’ (pattern [[Human]] *finisce* [[Event]]) and sense 2 ‘run out of, consume’ (pattern [[Human]] *finisce* [[Artefact]]) we find that the noun *cigarette* can appear in both sets, once interpreted as [[Event]] (smoking a cigarette) and once as [[Artefact]] (run out of cigarettes);[10]
- compare {V, S, T, A} with {T, A}, that is compare a given lexical set with the sets of all elements that have been defined as belonging to type T, filling argument position A; for example generate the lexical set of all [[Physical Object]]s, independently from the verb, to find out which is the core of common elements for this type;
- query for a specific T and extract all {V, S, T, A} in which it is instantiated; for example find all patterns in which an argument position has been filled by [[Human]], to verify how many verbs have a human subject or object;
- query for a specific L and extract all {V, S, T, A} in which it is instantiated; for example find all patterns in which an argument position has been filled by *libro* ‘book’ to verify in which situations books are likely to be involved (and which different aspects of books are highlighted).

In its current form, the Pattern Dictionary allows us to see the STs for each pattern, but only indirectly allows us to access the list of tokens that instantiate them in the corpus, i.e. by retrieving the corpus instances associated with the pattern and by checking them manually. This is so because in the original architecture each instance of the verb in the sample is assigned the code of a pattern as a whole.[11] Yet given this procedure, the resulting resource only permits one to perform a subset of the wished for queries, namely the extraction of:

- a V with all of its Ss;
- a V with a specific S;
- all Vs with a given T in some of their patterns;
- a pattern with a sentence in the corpus.

If one wants to examine the interaction between LSs and STs systematically, this must be performed manually or using computational heuristics such as those described in Rumshisky (2008). It makes sense to imagine extending the CPA methodology in order to allow for the tagging and retrieval of instances for each argument of each pattern.

2.1. Annotating the patterns onto the instances

One can then imagine an annotation interface added to the current Pattern Dictionary platform where, for each corpus instance, the human annotator is given the pattern to which it has been assigned by the corpus analyst and the description/sense of this pattern, and is asked, for each argument in the pattern, to link it to the corresponding portion of text in the instance from the corpus. This could be done by using an online annotation interface that displays the instance and the pattern structure that have already been linked by the corpus analyst, and to allow to link strings in the instance to the relevant syntactic and semantic information. When an instance is entered, the system could then transform the information into XML tagging.

The work of the annotator is made easier by the preliminary work of the corpus analyst, since the annotator does not have to disambiguate the sense, but is already presented with the right pattern and the right argument types to recognize; yet it is not a straightforward process and some problems arise.

(i) Type mismatch

A recurring problem that the annotator faces is that of the LS being bigger than the ST specified in the pattern. The annotator will have to take care of type mismatch: given the verb *bere* ‘drink’ and the pattern [[Human]] *beve* [[Beverage]], the annotator may be presented with an instance such as “*I ragazzi hanno bevuto una pinta insieme*” ‘The boys drank a pint together’; this means that the corpus analyst felt that examples like this (and others presenting nouns like *bottiglia* ‘bottle’, *bicchiere* ‘glass’, etc.), which strictly speaking present the form of [[Human]] *beve* [[Measure Unit / Artefact]], do not represent a different sense of the verb *bere*, and thus do not need to be described as a different pattern, but can rather be seen as cases of type shifting (or coercion), licensed by some property of the [[Measure Unit / Artefact]].[12] From the point of view of the annotation, it is therefore necessary to mark the type mismatch, when present, by adding the type associated with the noun out of context. In the annotated corpus the cases of type mismatch will be retrievable by querying for instances that present at least one noun with a semantic type different from the type assigned by the pattern.[13]

The case of type mismatch is only one of the problems that the annotator of the Patternbank may encounter. From a first annotation exercise we performed on authentic instances of the Italian implementation of the Pattern Dictionary it is clear that other issues arise during pattern annotation that ask for a further implementation of the annotation scheme. The majority of these issues relate to the mismatch between semantics and syntax. In fact, the majority of corpus instances that we annotated display a syntactic structure that is different from the one prototypically displayed by the pattern they instantiate. Following, therefore, are some examples of the mismatches that we found while annotating the patterns.

(ii) Syntactic alternation

When defining the pattern, the corpus analyst associates with each argument a “prototypical” syntactic position – the one corresponding to the syntactic form of the template, usually the active form. This does not yet create a direct mapping onto the syntactic function of the instantiated argument in the instance, since this can present a different syntactic form while maintaining the same sense, as in the active-passive alternation. This means that an argument specified in the object position in the pattern can be instantiated in the subject position in the instance (as *lo spumante* ‘the champagne’ in “*Arriva lo spumante che viene bevuto da tutti i presenti*” lit. ‘The champagne arrives that is drunk by everybody present’).

(iii) Anaphora resolution

In many cases, the information that is normally conveyed by an argument may be expressed by pronominal anaphora, lexical anaphora or, especially for Italian, zero anaphora (“*ø Prese la tazza e ø <u>la</u> bevve in un sorso*” ‘He/she took the cup and drank it in one swallow’). Consider that when the corpus analyst assigns a sentence like “*<u>la</u> bevve in un sorso*” to a pattern rather than another one, he/she must have some insight from the near linguistic context. In the annotation phase, while keeping the link between the arguments and their syntactic pronominal instantiation, we would ideally like to be able to mark the link between the pronoun (or the zero anaphora) and its antecedent in the near context. It is on the full form, of course, that the type check (for possible mismatch cases) is performed; a neat way of dealing with near-context anaphora is crucial when building a representative corpus of real instances since the instances with some kind of near anaphoric relation build up the vast majority of cases.

(iv) Unexpressed arguments

This problem can be exemplified referring once again to the case of passives, where the agent of the activity is usually omitted, and to zero anaphora, where the unexpressed argument usually refers to an entity that has been previously mentioned. Yet another example of unexpressed argument is provided by arguments that are left out because, in Fillmore's terms, "the identity of the referent that they instantiate is unknown or a matter of indifference" (1986: 96), for example "*Il vecchio si riposò, ø bevve ø, e fu assalito da uno strano pensiero*" 'The old man rested, drank, and was assailed by a strange thought'. In this case the annotator, by leaving one of the pattern arguments empty in some of the instances, will indirectly add important information to the pattern, namely that concerning the (contextually) obligatory and the droppable arguments, thus providing relevant information for the linguistic study of omissibility conditions in argument realization.

It does not seem plausible to solve problems such as (ii) by multiplying the patterns, since the alternation follows from a general rule and does not involve a change in meaning in V. This means that the instances displaying syntactic mismatch need to be kept within the same pattern, if they instantiate the same sense of a verb, and the mismatch needs to be taken care of in the mapping.[14]

2.2. Annotation architecture

From the analysis of the problems above, we propose to separate the syntax from the semantic level of annotation. Both the syntactic level and the semantic level are in turn divided into three layers. The semantic layers are:

- the **Pattern Type**, which records the semantic types that are imposed by the pattern for each position;

- the **Semantic Argument Filler**, which contains the lexical material that actually instantiates the semantic position in the instance;
- the **Instance Type** layer, which needs to be added when the Semantic Argument Filler instantiates a type that does not match with the Pattern Type, otherwise it is inherited from the Pattern.

The syntax layers mirror the semantic ones, and are:

- the **Pattern Role**, which contains the syntactic roles that are imposed by the pattern for each position;
- the **Syntactic Argument Filler**, which contains the lexical material that actually instantiates the syntactic position in the instance;
- the **Instance Role** layer, which needs to be added when the Syntactic Argument Filler instantiates a role that does not match with the Pattern Role, otherwise it is inherited from the Pattern.

The most important feature of this annotation scheme is that it records mismatches at different levels, i.e.:

- between the semantic type of the pattern and that of the instance, as happens in cases of coercion;
- between the syntactic role of the pattern and that of the instance, as happens with passives;
- between the argument filler of the semantic type and the argument filler of the syntactic role, as happens when the verb's syntactic positions are filled by anaphoric pronouns which refer back to a full form instantiating the semantic argument;
- between the number of arguments in the pattern and the number of arguments in the instance, as it happens when one argument is either dropped or added in the instance.

In the annotation architecture the different layers are represented by different levels of XML annotation, with indexes that maintain the co-reference of a given position on each layer of annotation.[15] We use an annotation scheme that is derived and adapted from the GLML (Generative Lexicon Markup Language) standard (Pustejovsky *et al*. 2008).[16]

The human annotator annotates the instances using a graphical interface. The different layers of annotation are presented as different fields to be filled. When an instance to be annotated is loaded into the interface, the related pattern structure (typed argument structure with the associated sense) is loaded from the pattern repository module of the resource. A link allows the annotator to access the ontology module in order to retrieve the instance type in the cases of type mismatch.[17]

2.3. Annotation examples

In this section, we give some examples to illustrate how the pattern annotation might be carried out by the annotator, and what the resulting XML code might look like. Some caveats are necessary here:

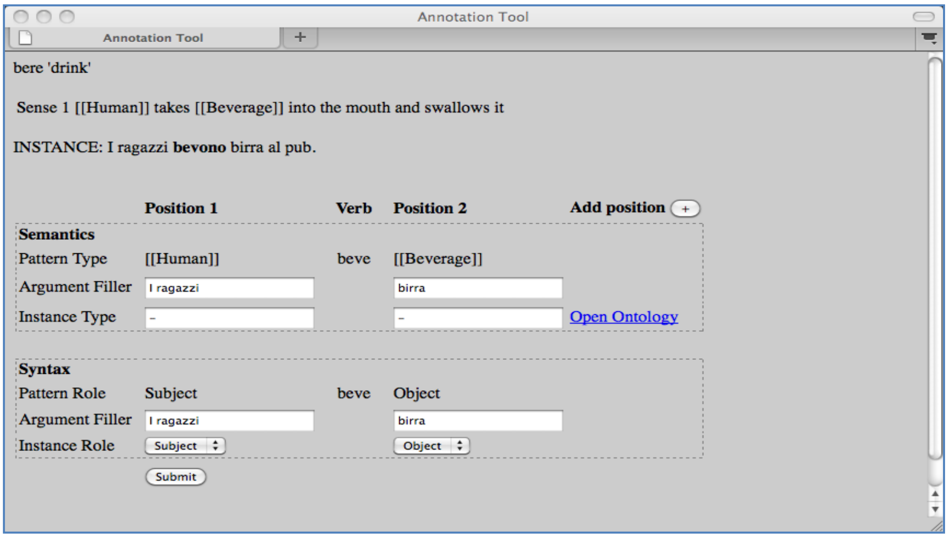
- the annotation task is under construction and the proposed version may not be the most usable one;
- some of the proposed layers could be treated in a semi-automatic way. In particular parsing and anaphora resolution algorithms could automatically take care of some of the layers, or at least suggest possible solutions to be amended by the human annotator;
- the large majority of real instances display mismatch onto several layers. This does not create problems for the architecture, yet for the sake of clarity we searched for examples which focus on one problem at a time. It is quite significant that the simplest cases needed to be constructed by modifying more complex real sentences. All examples are taken form it-WaC.[18]

The Pattern Repository contains the following pattern for the verb *bere* ‘to drink’, and has the following logical structure:

```
<pattern pattern_id=p15 lemma=bere sense=“take a drink into
   the mouth and swallow it”>
   <argument id=a1 pattern_sem_type=HUMAN pattern_syn_role=subj>
   <argument id=a2 pattern_sem_type=BEVERAGE pattern_syn_
   role=obj>
</pattern>
```

The pattern is defined as a combination of a given lemma with a sense. In the final architecture the corpus-driven types for each argument are stored in the Ontology module, and indexed in the pattern with an ID. Here both pattern and instance types are given in full form for clarity.

Example 1: No mismatch
"*I ragazzi **bevono** birra al pub*" / 'The boys drink beer at the pub'



bere 'drink'

Sense 1 [[Human]] takes [[Beverage]] into the mouth and swallows it

INSTANCE: I ragazzi **bevono** birra al pub.

	Position 1	Verb	Position 2	Add position +
Semantics				
Pattern Type	[[Human]]	beve	[[Beverage]]	
Argument Filler	I ragazzi		birra	
Instance Type	-		-	Open Ontology
Syntax				
Pattern Role	Subject	beve	Object	
Argument Filler	I ragazzi		birra	
Instance Role	Subject		Object	

Submit

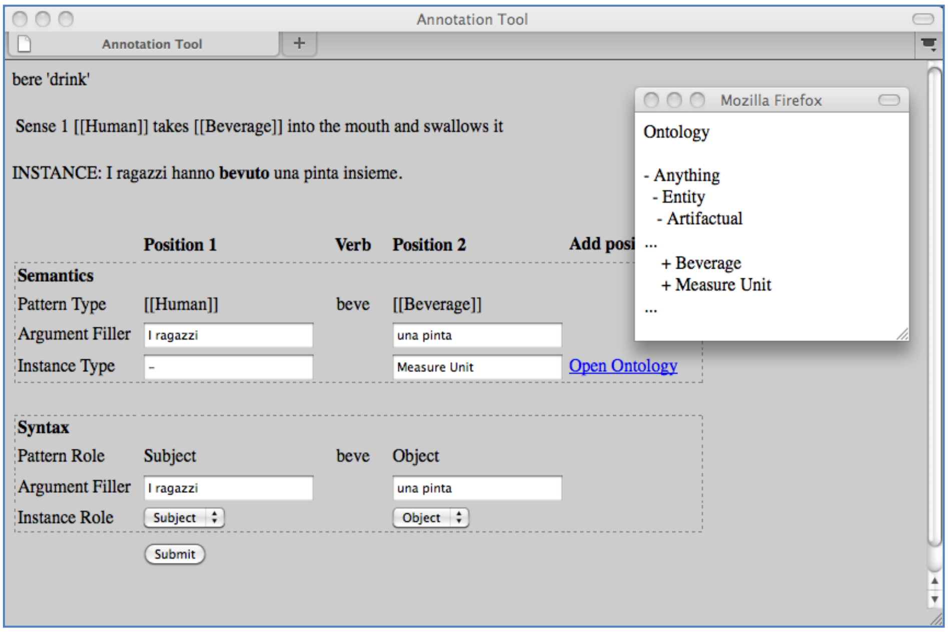
```
<instance tid=101>
<argument id=a1 pattern_id=p15 instance_sem_type=HUMAN in-
   stance_syn_role=subj>I ragazzi </argument> <verb pattern_id
   =p15>bevono </verb> <argument id=a2 pattern_id=p15 in-
   stance_sem_type=BEVERAGE instance_syn_role=obj>birra
   </argument>al pub.
</instance>
```

Here the mapping of the pattern onto the instance is very simple, since:

- the syntactic structure of the instance mirrors the prototypical syntactic structure of the pattern;
- the syntactic argument filler for each position corresponds to the semantic argument filler;
- there is no type mismatch and the type of each argument of the instance corresponds to the type selected by the pattern.

In the XML each element in the pattern is related to its instantiation via co-indexing: the verb has the same ID as the whole pattern and each argument has the same ID as the NP it refers to.[19] We speak here of NP and not noun, since we believe that it is the whole NP and not the noun that fills the argument position. The absence of mismatch can be derived from the fact that the attributes of each argument in the pattern are equivalent to those in the instance.

Example 2: Type mismatch
"*I ragazzi hanno* ***bevuto*** *una pinta insieme*"
'The boys drank a pint together'



bere 'drink'

Sense 1 [[Human]] takes [[Beverage]] into the mouth and swallows it

INSTANCE: I ragazzi hanno **bevuto** una pinta insieme.

	Position 1	Verb	Position 2	Add posi
Semantics				
Pattern Type	[[Human]]	beve	[[Beverage]]	
Argument Filler	I ragazzi		una pinta	
Instance Type	-		Measure Unit	
Syntax				
Pattern Role	Subject	beve	Object	
Argument Filler	I ragazzi		una pinta	
Instance Role	Subject		Object	

Ontology

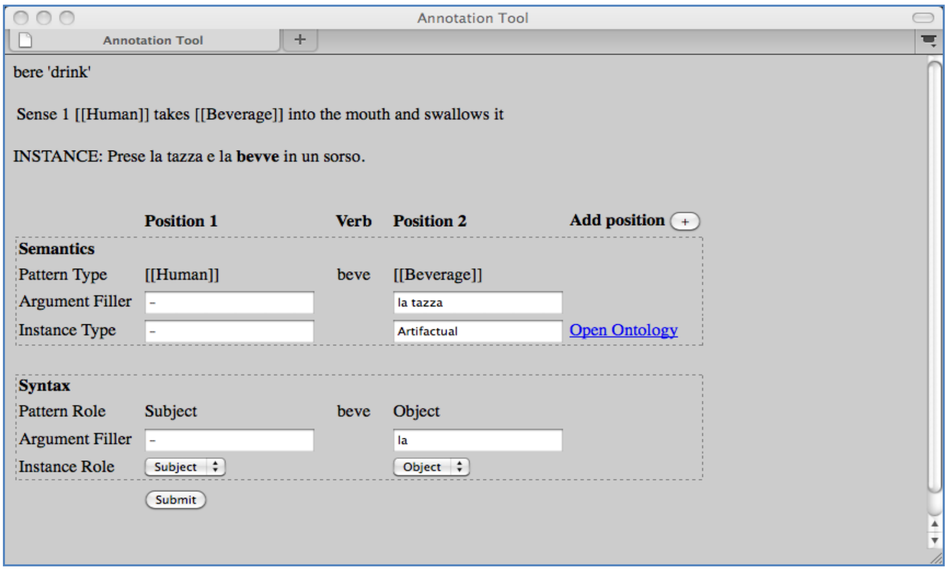
- Anything
 - Entity
  - Artifactual
...
  + Beverage
  + Measure Unit
...

```
<instance tid=102>
<argument id=a1 pattern_id=p15 instance_sem_type=HUMAN in-
   stance_syn_role=subj>I ragazzi </argument> <verb pat-
   tern_id=p15>hanno bevuto </verb> <argument id=a2 pat-
   tern_id=p15 instance_sem_type=MEASURE_UNIT in-
   stance_syn_role=obj>una pinta </argument>insieme.
</instance>
```

In this case:

- the syntactic structure of the instance mirrors the prototypical syntactic structure of the pattern;
- the syntactic argument fillers for each position correspond to the semantic argument fillers;
- *but* there is type mismatch on the direct object; this becomes evident when comparing the pattern and the instance semantic type of the second argument (argument id=a2).

Example 3: Anaphora
"*Prese la tazza e la **bevve** in un sorso*"
'He took the cup and drank it in one swallow'



bere 'drink'

Sense 1 [[Human]] takes [[Beverage]] into the mouth and swallows it

INSTANCE: Prese la tazza e la **bevve** in un sorso.

	Position 1	Verb	Position 2	Add position +
Semantics				
Pattern Type	[[Human]]	beve	[[Beverage]]	
Argument Filler	-		la tazza	
Instance Type	-		Artifactual	Open Ontology
Syntax				
Pattern Role	Subject	beve	Object	
Argument Filler	-		la	
Instance Role	Subject		Object	

```
<instance tid=103>
Prese <antecedent id=a2 pattern_id=p15 in-
   stance_sem_type=MEASURE_UNIT>la tazza </antecedent>e <argu-
   ment id=a1 pattern_id=p15 instance_syn_role=subj/> <argument
   id=a2 pattern_id=p15 instance_syn_role=obj>la </argument>
   <verb pattern_id=p15>bevve </verb>in un sorso.
</instance>
```

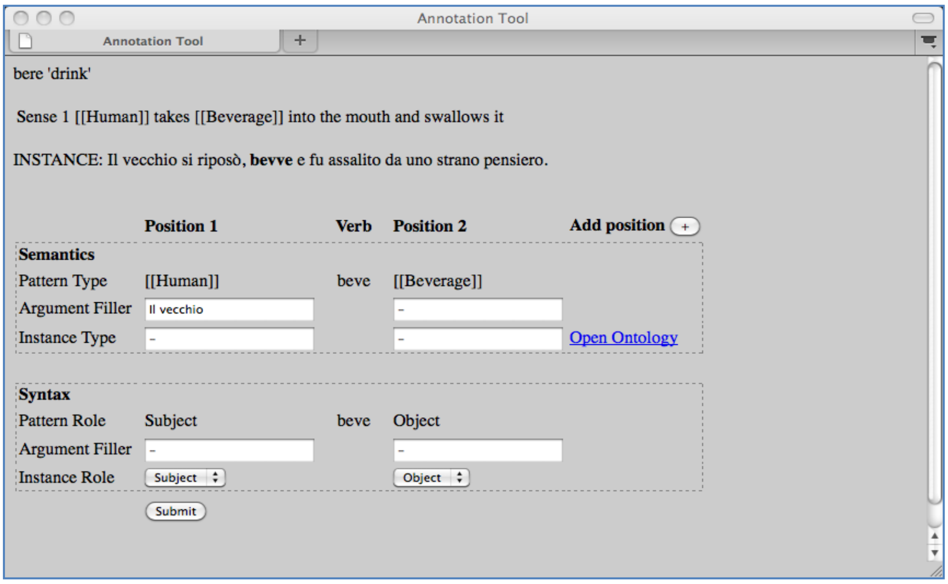
This case is more complex:

- the first argument position (argument id=a1) is present, but has no lexical material due to pro-drop; the tag for this argument is empty and there is no "instance_sem_type" (it simply inherits the one present in the pattern);
- due to the anaphora, one syntactic position of the verb is occupied by a pronoun, whereas the lexical filler of the semantic argument position must be found in the antecedent of this pronoun; the argument position "a2" is therefore split into two tags: "argument" bearing the syntactic information and "antecedent" bearing the semantic information; therefore both "argument id=a1" and "argument id=a2" lack semantic information, but the former has no antecedent in the instance, and must inherit its type from the pattern. The difference is substantial when one wants to build

lexical sets, that is the collection of all possible NPs that fill a certain position: in this instance only the second argument has an actual filler;

- just as in Example 2 there is type mismatch in argument “a2”.

Example 4: Unexpressed argument
“*Il vecchio si riposò, **bevve**, e fu assalito da uno strano pensiero*”
‘The old man rested, drank, and was assailed by a strange thought’



bere 'drink'

Sense 1 [[Human]] takes [[Beverage]] into the mouth and swallows it

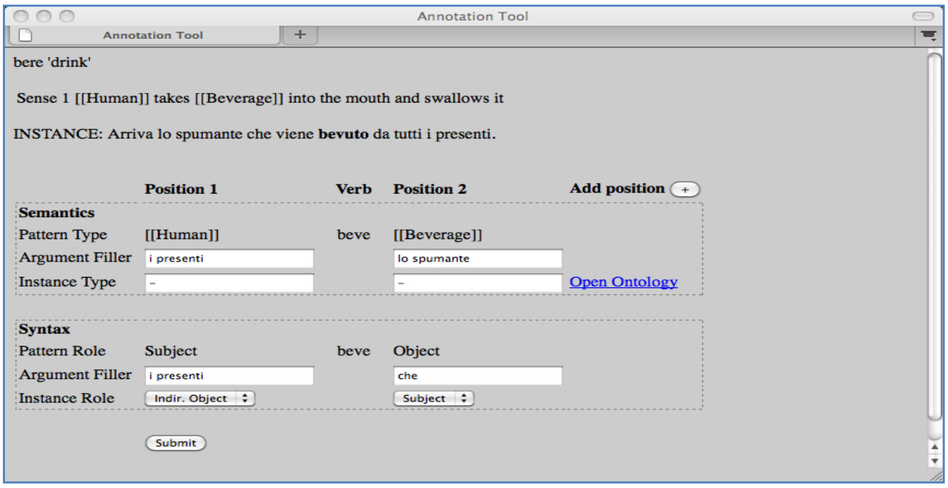
INSTANCE: Il vecchio si riposò, **bevve** e fu assalito da uno strano pensiero.

	Position 1	**Verb**	**Position 2**	**Add position** ⊕
Semantics				
Pattern Type	[[Human]]	beve	[[Beverage]]	
Argument Filler	Il vecchio		-	
Instance Type	-		-	Open Ontology
Syntax				
Pattern Role	Subject	beve	Object	
Argument Filler	-		-	
Instance Role	Subject		Object	

```
<instance tid=104>
<antecedent id=a1 pattern_id=p15 instance_sem_type=HUMAN>Il
   vecchio </antecedent>si riposò, <argument id=a1 pattern_id=p15 instance_syn_role=subj/> <verb pattern_id=p15>bevve </verb>e fu assalito da uno strano pensiero.
</instance>
```

In this case the first argument position is to be found in the antecedent to the dropped subject pronoun, which remains as an empty syntactic position. The second argument position, in contrast, is completely absent in the instance (no tag has “id=a2”); unexpressed arguments can be thus defined in that an argument which is present in the pattern is not co-referenced by any argument in the instance.

Example 5: Simple passive
"*Arriva lo spumante che viene* ***bevuto*** *da tutti i presenti*"
lit. 'The champagne arrives that is drunk by everybody'



bere 'drink'

Sense 1 [[Human]] takes [[Beverage]] into the mouth and swallows it

INSTANCE: Arriva lo spumante che viene **bevuto** da tutti i presenti.

	Position 1	Verb	Position 2	Add position +
Semantics				
Pattern Type	[[Human]]	beve	[[Beverage]]	
Argument Filler	i presenti		lo spumante	
Instance Type	–		–	Open Ontology
Syntax				
Pattern Role	Subject	beve	Object	
Argument Filler	i presenti		che	
Instance Role	Indir. Object		Subject	

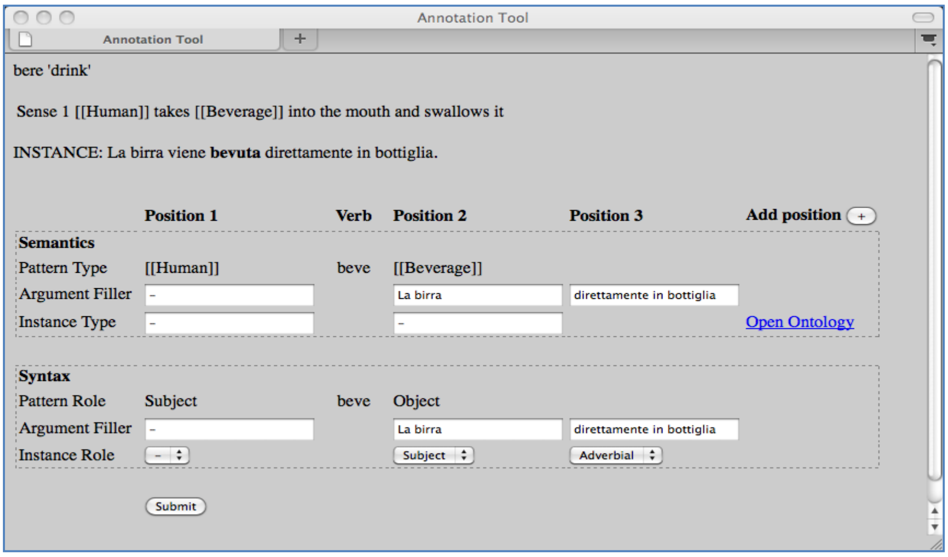
```
<instance tid=105>
Arriva <antecedent id=a2 pattern_id=p15 in-
   stance_sem_type=BEVERAGE>lo spumante </antecedent> <argument
   id=a2 pattern_id=p15 instance_syn_role=subj>che </argument>
   <verb pattern_id=p15>viene bevuto </verb> <argument id=a1
   pattern_id=p15 instance_sem_type=HUMAN in-
   stance_syn_role=IndObj>da tutti i presenti </argument>.
</instance>
```

In this case:

- the passive syntactic alternation causes the promotion of "argument id=a2" from object to subject, while "argument id=a1" becomes indirect object; therefore there is a mismatch between "pattern_syn_role" and "instance_syn_role" in both argument positions;
- the pronoun *che* fills the (shifted) syntactic role and is co-indexical with its antecedent which fills the semantic type of the argument.

Just as in the preceding examples, it is the indexing that allows us to link the argument position in the pattern with the corresponding one in the instance, since in some instances both "instance_syn_role" and "instance_sem_type" may not correspond to those in the pattern.

Example 6: Passive with obligatory adjunct
"*La birra viene **bevuta** direttamente in bottiglia*"
'Beer is drunk straight from the bottle'



bere 'drink'

Sense 1 [[Human]] takes [[Beverage]] into the mouth and swallows it

INSTANCE: La birra viene **bevuta** direttamente in bottiglia.

	Position 1	Verb	Position 2	Position 3	Add position +
Semantics					
Pattern Type	[[Human]]	beve	[[Beverage]]		
Argument Filler	-		La birra	direttamente in bottiglia	
Instance Type	-		-		Open Ontology
Syntax					
Pattern Role	Subject	beve	Object		
Argument Filler	-		La birra	direttamente in bottiglia	
Instance Role	-		Subject	Adverbial	

Submit

```
<instance tid=106>
<argument id=a2 pattern_id=p15 instance_sem_type=BEVERAGE in-
  stance_syn_role=subj>La birra </argument> <verb pat-
  tern_id=p15>viene bevuta </verb> <argument id=a3 pat-
  tern_id=p15 instance_syn_role=adv>direttamente in bottiglia
  </argument>.
</instance>
```

This is the most complex case because:

- the passive structure "argument id=a2" is promoted from the pattern syntactic position of object to the instance syntactic position of subject, and "argument id=a1" is unexpressed;
- the adverbial phrase is added to the argument structure of the instance, since its presence is required for pragmatic reasons as a result of agent omission;[20] a new argument is therefore present in the annotation; although both pattern and instance have two arguments, the mismatch is represented by the fact that there is no argument with "id=a3" in the pattern and no argument with "id=a1" in the instance.

In this proposal of annotation, phenomena like passivization, argument dropping, and argument promotion are treated in the same way as mismatch phenomena between semantics and syntax, and without any need of explicit coding. In order to extract a specific level of mismatch it is sufficient to run a query that, for each pattern and each argument position of each pattern, checks whether the relevant attribute in the instance corresponds to that in the pattern. Lexical sets are just as easy to create: in order to build the lexical set for position "a1" of pattern "p15" first select only the instances which contain a verb with "pattern id=p15" and then select the lexical material of "argument id=a2", where an attribute "instance_sem_type" is present; should the attribute "instance_sem_type" be absent, a tag "antecedent" is searched for; otherwise the instance does not contain a lexical filler for that position (as seen in Example 3).

3. What is the Patternbank useful for?

The resulting Patternbank may be useful for both linguistic theoretical studies and NLP applications. In the context of theoretical studies it may help researchers to investigate the relationship between lexical sets and semantic types by facilitating these tasks:

- cross-verb, cross-pattern, and cross-type comparison of lexical sets;
- extraction of the prototypical members for each lexical set with statistical measures;
- analysis of the distribution of certain argument selection phenomena with reference to specific subclasses of predicates;
- ideally, definition of rules that can predict, given a verb V with a set of senses {S1, S2, … Sn}, each having been defined with a typed argument structure, and given a noun N, with semantic type Tx, whether N can fill argument A1 of V or not, and if so, how this affects the meaning of V (for instance selecting S1 or S2) and the meaning of the N (for instance selecting Tx or alternatively coercing N into Ty).

Furthermore, the annotation of instances will help researchers to identify and extract phenomena lying on the interface between syntax and semantics, such as syntactic alternation, argument dropping and anaphora.

With respect to computational contexts, the Patternbank is planned as a reliable and carefully controlled corpus, that will not only be manually checked, but also checked for inter-annotator agreement, meant to be used as a domain-specific training set for developing applications among which are the following (in order of complexity):

- an application which, given a verb, a pattern, and an instance of the pattern, is capable of automatically mapping the argument structure onto the nouns in the instance, taking into account unexpressed arguments, type mismatch and anaphora;
- an application which, given a verb and its sense inventory, and a pattern for each sense, the types for each argument in those patterns, and a shimmering lexical set for each type in relation to the verb, can link a new occurrence of the verb to the correct sense, by reconstructing the argument structure and linking it to the most plausible pattern;
- an application which, given a verb with a certain syntactic structure, automatically maps each syntactic position to one or more semantic types, by comparing the lexical set selected by each argument by checking the types associated with other similar lexical sets derivable from known patterns in the Patternbank.

Finally, the Patternbank may be used to create a whole range of corpus-driven lexicographic products such as combinatorial dictionaries.

4. Further developments

The annotation system that we have outlined may be enriched with further information concerning the semantic features of the NP that fills a given argument position. More specifically, a further layer of information may be added both in the definition of the pattern and in the annotation of the instance; this level should contain semantic features that may be relevant in the definition and instantiation of the pattern, such as: number (singular/plural), mass/countable, and determinedness/non-determinedness.

The idea is that, just as the pattern selects some semantic type for the NP filling a certain argument position, it also selects certain semantic features for the NP. Just as it happens with type coercion, some NPs may have some features in isolation, which are somewhat subject to adjustment/coercion according to requirements of the pattern in which they are instantiated.

Acknowledgements

This paper is the result of cooperation between the co-authors initiated during the preparation of the Italian dataset for the SemEval-2010 task on *Argument Selection and Coercion*, organized by Brandeis University, Waltham, MA, together with ILC-CNR, Pisa. We would like to give credits to the work of researchers involved in the Corpus Pattern Analysis (CPA) project and in the Generative Lexicon Markup Language (GLML) effort, as well as to the organizers of the SemEval-2010 task 7.

Notes

[1] The repository we refer to is the Pattern Dictionary of English Verbs (PDEV), created using the Corpus Pattern Analysis (CPA) technique developed by Patrick and his collaborators (cf. Hanks 1996, Pustejovsky & Hanks 2001, Hanks 2004, Pustejovsky, Hanks & Rumshisky 2004, Hanks & Pustejovsky 2005, Hanks 2006, 2007, forthcoming, inter alia). A pattern, in the CPA sense, is a semantically motivated and recurrent piece of phraseology. Patterns for verbs consist of valencies plus semantic types of arguments within valencies – populated by lexical sets, i.e. paradigmatic lists of words that may fill each valency in the pattern. For example: [[Human]] attend [[Event]], where Lexical set [[Event]] = {meeting, conference, funeral, ceremony, course, school, seminar, dinner, reception, workshop, wedding, concert, premiere, …}.

[2] The convention of double square brackets with capital initial letters is used to indicate the names of semantic types.

[3] Hanks (1996: 82) introduces the distinction between *bona-fide* set members vs. *ad-hoc/honorary* set members.

[4] In semantic composition, the opposite phenomenon also occurs (the meaning of N modulates the meaning of V in context). We will simplify for the present discussion and assume that each V has a set of pre-defined senses, one of which is selected by N in composition. This allows us to focus on the problems that arise with regard to the modelling of noun ambiguity in context. It is evident, however, that in order to give a full account of syntagmatic processes occurring in semantic composition, both directions of modulation should be considered.

[5] Cf. Pustejovsky 1995, Copestake & Briscoe 1995, inter alia.

[6] Note that the problem we point out here is the tension between the requirements of a predication and the categories in our conceptual system. This interplay precedes operations of coercion that might take place in composition.

[7] In Generative Lexicon terms (GL, cf. Pustejovsky 1995) the Formal role is the dimension of meaning specifying the basic category that distinguishes an object within a larger domain, and allows one to answer the question: "What is it?" (a [[Person]], a [[Physical Object]], an [[Event]], an [[Animal]], a [[Substance]], etc.).

[8] This point is included in the architecture of the SIMPLE ontology (Lenci *et al*. 2000), where qualia roles are assumed as organizing principle.

[9] For example, in verb-argument composition, one might exploit the function or constitution of an object and not its Formal: cf. "She poured two glasses and gave him one", where *pour* coerces the artefact *glass* to the drink it typically contains (for a corpus-informed investigation of coercion mechanisms in texts based on GL theory, see Pustejovsky & Jezek 2008).

[10] Queries like this can be useful to understand how the senses of a single verb relate to each other: the fact that the senses 'bring to an end' and 'run out of' are expressed by the same verb *finire* is justified by the fact that most of the times if something runs out this happens because it has been previously consumed to its end (as with cigarettes).

[11] Briefly, in CPA each verb is analyzed according to the following procedure: first, a sample concordance for each target verb is created (250 hits); second, the semantic types of the argument fillers are examined and the typical syntagmatic patterns of the verb are identified (e.g. for *read*: [[Human]] read [[Document]]); third, each line of the sample is assigned to one of the drafted patterns; fourth, both the patterns and the associated concordances are stored in the pattern repository. The analysis of a verb in the Pattern Dictionary is considered complete when its patterns have been recognized, described and mapped onto the sample sentences of the corpus.

[12] In Generative Lexicon terms the licensing property is to be searched for in the qualia structure of the noun; in the case of *glass* it is the telic value "hold (liquid)" that is relevant.

[13] So far, an effort has been made to produce a corpus annotated for coercion operations (for both English and Italian) to be used as training set for the SemEval-2010 task 7 on *Argument Selection and Coercion*, using the Generative Lexicon Markup Language (GLML) annotation standard. The present contribution stems from this work and may be understood as a proposal to frame it within the Italian implementation of the Pattern Dictionary project.

[14] On the basis of the criterion "syntactic alternation within a sense = same pattern", the causative/inchoative alternation is not treated as a single pattern in the Pattern Dictionary, given that the two constructions exhibit two distinct

senses (*Mary breaks the window* = causative / *The window breaks* = non-causative).

[15] For example, the dropping of an argument in an instance is represented by the fact that there is a position in the pattern that is not co-indexed by any position in the semantic and syntactic layers of the instance.

[16] GLML is a mark-up language aimed at annotating compositional operations in natural language text based on the Generative Lexicon theory. It allows one to annotate both the semantic type assigned by predicates to their argument(s) (*target type* in GLML terms) and the surface type of the entities involved in argument selection (*source type*).

[17] The overall architecture of the Patternbank planned for Italian will consist of three main modules: (a) a repository of corpus-derived Italian verb patterns mapped onto corpus-derived verb meanings (the Pattern Dictionary of Italian Verbs, PDIV); (b) a corpus-driven shallow semantic type repository (the Italian Linguistic Ontology, ILO) containing semantic types of arguments that are relevant for distinguishing between different verb senses; and (c) a repository of annotated instances (the Italian Patternbank) where each predicate and its arguments are mapped onto the relevant pattern – this corpus will be linked to both (a) and (b).

[18] itWaC (Italian Web as Corpus) is a large tokenized and POS-tagged corpus of Italian texts that was prepared by Marco Baroni & Adam Kilgarriff in a web crawl as described at EACL 2006 (Baroni & Kilgarriff 2006; Dimensions: 2 billion words; Availability: uploaded in the Sketch Engine (licence needed)). itWaC is the corpus we use for the identification of the verb patterns in the Italian implementation of the Pattern Dictionary.

[19] Each argument also bears the reference to the pattern/verb it refers to; this is redundant in the given examples, but it is meant to allow for more than one pattern/verb to be annotated in one instance.

[20] With passives, as a result of agent omission, an obligatory adjunct often comes in to provide the focus that serves to convey new information in the discourse.

References

Baroni, M. & A. Kilgarriff. 2006. 'Large Linguistically-Processed Web Corpora for Multiple Languages' in *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento: ACL, 87–90.

Copestake, A. & T. Briscoe. 1995. ‘Semi-productive Polysemy and Sense Extension’. *Journal of Semantics* 12.1: 15–67.

Fillmore, C. J. 1986. ‘Pragmatically Controlled Zero Anaphora’. *Bulletin of the Linguistic Society* 12: 95–107.

Hanks, P. 1996. ‘Contextual Dependency and Lexical Sets’. *International Journal of Corpus Linguistics* 1.1: 75–98.

Hanks, P. 2004. ‘Corpus Pattern Analysis’ in G. Williams & S. Vessier (eds.). 2004. *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004, Lorient, France, July 6-10*, 2004. Lorient: Faculté des Lettres et des Sciences Humaines, Université de Bretagne Sud, 87–97.

Hanks, P. 2006. ‘The Organization of the Lexicon: Semantic Types and Lexical Sets’ in E. Corino, C. Marello & C. Onesti (eds.). 2006. *Atti del XII Congresso Internazionale di Lessicografia, Torino, 6-9 settembre 2006 / Proceedings XII Euralex International Congress, Torino, Italia, September 6th-9th, 2006*. Alessandria: Edizioni dell’Orso, 1165–1168.

Hanks, P. 2007. *Pattern Dictionary of English Verbs (PDEV) – Project Page*. Online at http://deb.fi.muni.cz/pdev/.

Hanks, P. (forthcoming). *Lexical Analysis: Norms and Exploitations*. Cambridge, MA: MIT Press.

Hanks, P. & E. Jezek. 2008. ‘Shimmering Lexical Sets’ in E. Bernal & J. DeCesaris (eds.). 2008. *Proceedings of the XIII EURALEX International Congress (Barcelona, 15-19 July 2008)* (Sèrie Activitats 20). Barcelona: Institut Universitari de Lingüística Aplicada, Universitat Pompeu Fabra, 391–402.

Hanks, P., E. Jezek & A. Lenci. 2008. ‘What is a Linguistic Ontology? Ontologies, Semantic Types and Combinatorial Constraints’. *Abstract submitted to Workshop 12: Linguistic Studies of Ontology, Held at the 18th International Congress of Linguists, Korea University, Seoul, Korea, July 21-26, 2008*.

Hanks, P., K. Pala & P. Rychlý. 2007. ‘Towards an Empirically Well-founded Semantic Ontology for NLP’ in *Proceedings of the 4th International Workshop on Generative Approaches to the Lexicon, Paris, May 10-11, 2007*.

Hanks, P. & J. Pustejovsky. 2005. ‘A Pattern Dictionary for Natural Language Processing’. *Revue française de linguistique appliquée* 10.2: 63–82.

Jackendoff, R. 2002. *Foundations of Language*. Oxford: Oxford University Press.

Lenci, A, N. Bel, F. Busa, N. Calzolari, E. Gola, M. Monachini, A. Ogonowski, I. Peters, W. Peters, N. Ruimy, M. Villegas & A. Zampolli. 2000. ‘SIMPLE: A General Framework for the Development of Multilingual Lexicons’. *International Journal of Lexicography* 13.4: 249–263.

Pustejovsky, J. 1995. *The Generative Lexicon*. Cambridge, MA: The MIT Press.

Pustejovsky, J. & P. Hanks. 2001. 'Very Large Lexical Databases'. Tutorial at ACL 2001, Toulouse, 6-11 July 2001. Tutorial notes available online at http://www.cs.brandeis.edu/~llc/publications/ACL-Toulouse-Slides.pdf.

Pustejovsky, J., P. Hanks & A. Rumshisky. 2004. 'Automated Induction of Sense in Context' in *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004), 23-27 August 2004*. Geneva: University of Geneva, 924–930.

Pustejovsky, J. & E. Jezek. 2008. 'Semantic Coercion in Language. Beyond Distributional Analysis'. *Italian Journal of Linguistics / Rivista Italiana di Linguistica* 20.1, 181–214.

Pustejovsky, J., A. Rumshisky, J. L. Moszkowicz & O. Batiukova. 2008. 'GLML: A Generative Lexicon Markup Language'. Manuscript presented at the GL 2008 workshop, Pisa, Istituto di Linguistica Computazionale di Pisa.

Rumshisky, A. 2008. 'Resolving Polysemy in Verbs: Contextualized Distributional Approach to Argument Semantics'. *Italian Journal of Linguistics / Rivista Italiana di Linguistica* 20.1, 215–240.